



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/900,060	07/06/2001	Travis J. Muhlestein	MSFT115921	7821
26389 7590 01/03/2007 CHRISTENSEN, O'CONNOR, JOHNSON, KINDNESS, PLLC 1420 FIFTH AVENUE SUITE 2800 SEATTLE, WA 98101-2347			EXAMINER VU, TUAN A	
			ART UNIT 2193	PAPER NUMBER
SHORTENED STATUTORY PERIOD OF RESPONSE		MAIL DATE	DELIVERY MODE	
3 MONTHS		01/03/2007	PAPER	

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No.	Applicant(s)	
	09/900,060	MUHLESTEIN ET AL.	
	Examiner	Art Unit	
	Tuan A. Vu	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 23 October 2006.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1 and 3-17 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1, 3-17 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 10/23/06.

As indicated in Applicant's response, claims 1, 7-9, 17 have been amended. Claims 1, 3-17 are pending in the office action.

Claim Rejections - 35 USC § 101

2. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

3. Claims 8, 17 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

The Federal Circuit has recently applied the practical application test in determining whether the claimed subject matter is statutory under 35 U.S.C. § 101. The practical application test requires that a "useful, concrete, and tangible result" be accomplished. An "abstract idea" when practically applied is eligible for a patent. As a consequence, an invention, which is eligible for patenting under 35 U.S.C. § 101, is in the "useful arts" when it is a machine, manufacture, process or composition of matter, which produces a concrete, tangible, and useful result. The test for practical application is thus to determine whether the claimed invention produces a "useful, concrete and tangible result".

As set forth in the previous Office Action, these claims have been objected to because they contain language that would render the invention non-statutory (with emphasis added). As amended (emphasis added), these claims remain deficient for now they still recite 'computer-controlled apparatus' without explicit support as to what exactly this apparatus amounts to (or is constituted of) in terms of tangible hardware embodiment. The claims 1 and 9 on which these claims depend only recite a method with a *runtime* execution *engine*. In the Specifications, there is not sufficient teaching that this engine is a hardware engine; therefore this engine is construed as a .NET or Windows non-hardware engine to execute software instructions. Software functionality in the absence of hardware storage or tangible embodiment or execution engine

Art Unit: 2193

remains functionality not capable of being realized into real-world tangible result, i.e. an abstract idea. Lacking a hardware support in order to realize the functionality of the method of the base claims, claims 8 and 17 amount to non-practical application; and are rejected for leading to a non-statutory subject matter.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

5. Claims 1, 3-17 rejected under 35 U.S.C. 103(a) as being unpatentable over Microsoft Corporation, "Microsoft Windows Management Instrumentation Scripting", April 1999, pp. 1-15 (hereinafter MSWMI), further in view of Admitted Prior Art (APA – see BACKGROUND of application).

As per claim 1, MSWMI discloses a computer-implemented method for providing access to instrumentation data from within a managed code runtime environment, the method comprising

providing an application (e.g. WMI technology – Introduction) from in a runtime-aware programming language (e.g. Introduction: *enterprise environment, model* - pg 1; Object, Information Model: pg. 5-11), the application being suitable for execution by a runtime engine in a runtime environment (Note: application of a enterprise application with modeling and interface or extension APIs reads on application with a runtime-aware programming language);

Art Unit: 2193

executing the application in a runtime environment having a runtime engine, wherein there is a defined contract of operation between the executing application and the runtime engine to delegate certain application tasks to the runtime engine that enable the runtime engine to service requests (e.g. **Windows Management Instrumentation Technology: Access to monitor, command, control any entity...underlying mechanism, API ... Interoperability ...providing and accessing management ...extend the information ...connect one or more sources of management information ...capture instrumentation, detailed queries** --pg. 1, bottom to pg. 2 , top) from the executing application at runtime;

including requests for instrumentation data representing management information about other applications and devices available outside the runtime environment (e.g. *to capture instrumentation data from device drivers kernel* ..- pg. 2, 5th bullet-top; *Performance Monitor Provider* – pg. 4, 4th bullet; **WMI Architecture Overview: using WMI APIs ... providers supply ... CIM object Manager with data from managed objects, handle requests ; interface between management applications and data providers ... common programming interface to Windows Management Instrumentation,**– pg.3, middle; Fig. 1, pg. 4; *WMI Providers data that is not available from the CIMOM ... forward to WMI Provider data and event notifications for managed objects* – top pg. 4; **Advantages of Using WMI Scripting: custom providers can ... cover vendor specific instrumentation (for system, applications, devices...), Extensible Providers instrumentation** – 3rd bullet, pg. 5);

receiving a request at the runtime engine from the executing application for instrumentation data available outside said runtime environments the request including

Art Unit: 2193

a path of an instrumentation data object (e.g. *SWbemObjectPath* – pg. 6, Features: Object Creation; *SWbemObjectPath* – bottom, pg. 7) for accessing the instrumentation data (e.g. pg. 2, 5th bullet-top; pg.3, middle; Fig. 1, pg. 4; top pg. 4),

options used to retrieve (e.g. *SWbemServices Object: Get, Delete, InstancesOf, ExecQuery, AssociatorsOf ...* pg. 7, middle; *GetObjectText_, SpawnInstance_*, pg. 9, middle) the instrumentation data object, and

an identification of a parent (e.g. *ParentNameSpace*, pg. 8, 3rd bullet)of the instrumentation data object;

transmitting a corresponding request for said instrumentation data to an instrumentation data source external to said runtime environment, receiving a response to said corresponding request from said instrumentation data source (e.g. *to capture instrumentation data from device drivers kernel ...* - pg. 2-top, 5th bullet; **WMI Architecture Overview: using WMI APIs ... providers supply ... CIM object Manager with data from managed objects, handle requests ; interface between management applications and data providers ... common programming interface to Windows Management Instrumentation,** – pg.3, middle; Fig. 1, pg. 4; *WMI Providers data that is not available from the CIMOM ... forward to WMI Provider data and event notifications for managed objects* – top pg. 4);

converting said response to a format that is compatible with said runtime environment (**Windows Management Instrumentation Technology: supports the syntax of CIM, MOF, common programming interface, scripting support** - pg. 1, bottom – Note: WMI environment working in conjunction with providers via scripting, and API for retrieval of remote objects, while supporting syntax of all interfaces reads on converting to syntax compatible for the WMI);

Art Unit: 2193

responding to said request for instrumentation data with said converted response (Note: request for data using API and collecting data into a compatible form for the modeling/instrumentation application reads on responding to request for such instrumentation data).

MSWMI does not explicitly disclose that the application for the runtime-aware language is written in an intermediate language, nor does MSWMI disclose that the runtime engine to execute said application is configured to execute such intermediate language. The use of WMI (Microsoft WMI or MSWMI) in an application environment known as .NET platform has been well-established at the time the invention was made as set forth in APA (see BACKGROUND of Application: pg. 3, bottom para; pg. 4, top two para), according to which Microsoft .NET platform utilizes Microsoft WMI to effect the interface necessary to retrieve instrumentation data which is taught by MSWMI to the .NET platform, wherein .NET application is compiled as intermediate code (IL) so that the IL is admittedly being run using by a Microsoft .NET runtime engine (APA, see BACKGROUND: pg. 2, 3rd para). Based on MSWMI being also a Microsoft product used in retrieving instrumentation data for Microsoft runtime application, it would have been obvious for one of ordinary skill in the art at the time the invention was made to utilize the WMI (as by MSWMI) so that it supports an interface to an application written in IL and executed by a .NET runtime engine (as by APA) because according to the Microsoft and APA, .NET applications programs are platform independent designed to communicate with many other sources, and since MSWMI is also a product of Microsoft running as an interface in its own form in tandem with the Microsoft .NET environment (see APA, pg. 3-4) for rendering a variety of services to retrieve such multi-source data for the managed code of .NET (see APA), using the

Art Unit: 2193

WMI into support a Microsoft .NET application as set forth by APA would be the very purpose of WMI (see APA: BACKGROUND, pg. 4) in light of .NET methodology's endeavor to obtain instrumentation data as purported by MSWMI.

As per claim 3, MSWMI discloses converting instrumentation data object to a management object that is compatible with said runtime environment (see claim 1; *Using WMI technology .. create ...applications that implement ... features such as displaying system information, generating ... inventory resources...processing events* – pg. 3, Management Applications, bottom – Note: integrating data from request via API calls in order to integrate them for display in application via processing therein reads on converting requested data in runtime compatible form).

As per claim 4, MSWMI discloses wherein said management object encapsulates properties of the instrumentation data object (e.g. Standard inheritable methods – pg. 3, top, 2nd bullet; **Features: Monikers, for encapsulating the location** - pg 6, middle) accessible through said instrumentation data source, including

properties representing the path (e.g. Features: Object Creation, pg. 6; SWbemObjectPath – bottom, pg. 7) of the instrumentation data object for accessing the instrumentation data,

the options used to retrieve (e.g. *SWbemServices Object: Get, Delete, InstancesOf, ExecQuery, AssociatorsOf* ... pg. 7, middle) the instrumentation data object and

the identification of the parent (e.g. *ParentNameSpace*, pg. 8, 3rd bullet) of the instrumentation data object.

As per claims 5-6, MSWMI discloses wherein said response comprises an indication that an operation was unsuccessful and wherein converting said response to said format comprises

Art Unit: 2193

generating a management exception object; said indication that an operation was successful comprises error codes (e.g. **Advantage of Using WMI Scripting**: 4th bullet: *built-in features ... exception* --pg. 5, middle; **Features: Error Handling** - pg 6, middle; **Asynchronous example: HRESULT, ErrorObject** – pg. 14, 2nd para; **SwbemLastError object: read-once semantics... cleared after reading** – pg. 9, bottom).

As per claim 7, MSWMI discloses a computer-readable medium comprising instructions which, when executed by a computer, cause the computer to perform the method of any one of claims 1 and 3-6 (e.g. Note: a computer system capable of supporting script, encapsulating of objects, API calls, binding object-oriented instances to a model, and display of instrumentation data or event processing as in claims 1, 3-6 reads on inherent computer readable medium for storing such software capabilities).

As per claim 8, MSWMI discloses a computer-controlled apparatus for performing the method of any one of claims 1 and 3-6 (see claim 7).

As per claim 9, MSWMI discloses a computer-implemented method for accessing instrumentation data from within a runtime environment, wherein the runtime environment provides a runtime engine that executes an application compiled in a runtime-aware language (e.g. Introduction: *enterprise environment, model* - pg 1; Object, Information Model: pg. 5-11-- Note: application of a enterprise application with modeling and interface or extension APIs reads on application with a runtime-aware programming language), the method comprising:

receiving a request from the application for instrumentation data representing management information about other applications and devices available outside the runtime environment (*to capture instrumentation data from device drivers kernel* ..- pg. 2-top, 5th bullet; **WMI**

Art Unit: 2193

Architecture Overview: *using WMI APIs ... providers supply ... CIM object Manager with data from managed objects, handle requests ; interface between management applications and data providers ... common programming interface to Windows Management Instrumentation, – pg.3, middle; Fig. 1, pg. 4; WMI Providers data that is not available from the CIMOM ... forward to WMI Provider data and event notifications for managed objects – top pg. 4,*

the request comprising a path of an instrumentation data object for accessing said instrumentation data (e.g. Features: Object Creation, pg. 6; SWbemObjectPath – bottom, pg. 7), options used to retrieve (e.g. *SWbemServices Object: Get, Delete, InstancesOf, ExecQuery, AssociatorsOf ... pg. 7, middle; GetObjectText_, SpawnInstance_, pg. 9, middle*) the instrumentation data objects and a namespace (e.g. **SWbemServices object: object ...connection to a namespace – pg. 7, middle; ParentNameSpace, pg. 8, 3rd bullet**) of the instrumentation data object;

in response to said request, querying for said instrumentation data, using the path to said instrumentation data object for accessing said instrumentation data; determining whether said instrumentation data was successfully returned (**WMI Scripts Usage: Method Execution, Queries, remote Access, pg. 11; Asynchronous example: HRESULT, ErrorObject – pg. 14, 2nd para – Note: scripting with path parameters reads on using path to incorporate in the query effected via API calls**); and

in response to determining that said instrumentation data was successfully returned, constructing said management object in the runtime environment and populating said management object (e.g. CIM Object Collection-*SwbemObjectSet*, pg 11 – Note: object set after collecting of data from remote access reads on populating CIM model; **Features: Object**

Art Unit: 2193

Creation, Collections, Direct Access, pg. 6; SwbemEventSource Object, SwbemNamedValueSet collection, SwbemObject) with said instrumentation data.

MSWMI does not explicitly disclose that the application for the runtime-aware language is written in an intermediate language. But this limitation has been addressed in claim 1 above.

As per claim 10, MSWMI discloses wherein constructing said management object in the runtime environment and populating said management object with said instrumentation data includes binding an instance of a management object class (e.g. **Features: Monikers** – pg. 6, middle) to said instrumentation data object for accessing said instrumentation data source.

As per claim 11, MSWMI discloses constructing a management scope object identifying the namespace (**SWbemServices object: object ...connection to a namespace** – pg. 7, middle; *ParentNameSpace*, pg. 8, 3rd bullet) associated with said instrumentation data object for accessing said instrumentation data.

As per claims 12-13, MSWMI discloses constructing a management path object identifying the path (**Features: Object Creation**, pg. 6; *SWbemObjectPath* – bottom, pg. 7), and specifying the options to retrieve (e.g. *SWbemServices Object: Get, Delete, InstancesOf, ExecQuery, AssociatorsOf ...* pg. 7, middle; *GetObjectText_, SpawnInstance_*, pg. 9, middle) said instrumentation data object for accessing said instrumentation data.

As per claim 14, MSWMI discloses throwing a management exception object (**Advantage of Using WMI Scripting**: 4th bullet: *built-in features ... exception* --pg. 5, middle; **Features: Error Handling** - pg 6, middle; **Asynchronous example: HRESULT, ErrorObject** – pg. 14, 2nd para; **SwbemLastError object: read-once semantics... cleared after reading** – pg. 9, bottom) in response to determining that said instrumentation data was not successfully returned.

As per claim 15, MSWMI discloses wherein properties of said management object may be accessed utilizing an indexer (e.g. *SwbemNamedValueSet*: ...*indexing mechanism* – *SwbemNamedValueSet* collection, pg. 8).

As per claims 16-17, MSWMI discloses a computer-readable medium and computer-controlled apparatus for performing the method of any one of Claims 9-15 (refer to claims 7-8).

Response to Arguments

6. Applicant's arguments filed 10/23/06 have been fully considered but they are not persuasive. Following are the Examiner's observation in regard thereto.

(A) Applicants have submitted that MWMIS describes using script language and that scripts themselves cannot be viewed as 'application compiled into ... intermediate form from ... programming language'; nor can they be viewed as requiring an engine 'configured to execute ... intermediate form' (see Appl. Rmrks, pg. 7). First, the claim mentions about an application when run enable services via delegation of tasks for requests for instrumentation data, the implementation of the so-recited *instrumentation data* request is claimed as a sequence of transmitting or receiving. The applicants appear to have analogized the API or class objects (to retrieve data) to the very environment in which interfaces are called upon to provide this instrumentation data. The rejection has set it clear that an application is an application; an interface is an interface, and a request is being implemented via such interface call. The rejection does not equate any script to an application environment, because it is set clear in MSWMI that a runtime environment might request instrumentation data via using application interface; and scripting is but one among many approaches for implementation this interfacing code; and this is further evidenced in APA, according to which the .NET environment remains

Art Unit: 2193

the runtime for which data needs to be retrieved using interface calls; and this is what WMI is for, or what MSWMI is purported to achieve. The MSWMI does not mix up script with a runtime application because script is but an form of implementation for instrumentation whereas a runtime in Microsoft has its variety of executable form, one of which is the .NET (see APA) intermediate format. The argument is largely misplaced and not convincing.

Besides, the *intermediate form* limitation is a newly added feature for the present submission; and a new rejection has been set forth accordingly, rendering the argument moot because it does not apply to a Previous state of an Office Action according to 37 CFR §1.111(b), whereby the argument has to point to deficiency of the very rejection pertinent to the corresponding set of claimed subject matter, which is not the case here.

(B) The argument against claim 9 is also falling under the ambit of the newly added limitation, hence will be referred to section A above; i.e. not persuasive and not conforming to prima facie required by 37 CFR §1.111(b).

The Arguments are in all not appropriate to overcome the state of the rejection; the claims are thus rejected as set forth above.

Conclusion

7. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after

Art Unit: 2193

the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (272) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571)272-3756.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence - please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Application/Control Number: 09/900,060

Page 14

Art Unit: 2193

A handwritten signature in black ink, appearing to read 'Tuan A Vu', with a long horizontal flourish extending to the right.

Tuan A Vu
Patent Examiner,
Art Unit 2193
December 27, 2006